



Mol-CycleGAN - A Generative Model for Molecular Optimization

Łukasz Maziarka^{1,2(✉)}, Agnieszka Pocha², Jan Kaczmarczyk¹,
Krzysztof Rataj¹, and Michał Warchol¹

¹ Ardigen, Kraków, Poland

lukasz.maziarka@ardigen.com

² Jagiellonian University, Kraków, Poland

Abstract. During the drug design process, one must develop a molecule, which structure satisfies a number of physicochemical properties. To improve this process, we introduce Mol-CycleGAN – a CycleGAN-based model that generates compounds optimized for a selected property, while aiming to retain the already optimized ones. In the task of constrained optimization of penalized logP of drug-like molecules our model significantly outperforms previous results.

Keywords: Drug design · Generative models · Molecular optimization

1 Introduction

The principal goal of the drug design process is to find new chemical compounds that are able to modulate the activity of a given target in a desired way [13]. However, finding such molecules in the high-dimensional chemical space of all molecules without any prior knowledge is nearly impossible. *In silico* methods have been introduced to leverage the existing knowledge, thus forming a new branch of science - computer-aided drug design (CADD) [1, 12].

The recent advancements in deep learning have encouraged its application in CADD [4]. One of the main approaches is *de novo design*, that is using generative models to propose new molecules that are likely to possess the desired properties [3, 5, 15, 17].

In the center of our interest are the hit-to-lead and lead optimization phases of the compound design process. Their goal is to optimize the drug candidates identified in the previous steps in terms of the desired activity profile and their physicochemical and pharmacokinetic properties.

To address this problem, we introduce **Mol-CycleGAN** – a generative model based on CycleGAN [19]. Given a starting molecule, it generates a structurally similar one but with a desired characteristic. We show that our model generates molecules that possess desired properties while retaining their structural similarity to the starting compound. Moreover, thanks to employing graph-based representation, our algorithm always returns valid compounds.

To assess the model’s utility for compound design we evaluate its ability to maximize penalized logP property. Penalized logP is chosen because it is often selected as a testing ground for molecule optimization models [7, 18], due to its relevance in the drug design process. In the optimization of penalized logP for drug-like molecules our model significantly outperforms previous results. To the best of our knowledge, Mol-CycleGAN is the first approach to molecule generation that uses the CycleGAN architecture.

2 Mol-CycleGAN

Mol-CycleGAN is a novel method of performing compound optimization by learning from the *sets* of molecules with and without the desired molecular property (denoted by the sets X and Y). Our approach is to train a model to perform the transformation $G : X \rightarrow Y$ (and $F : Y \rightarrow X$) which returns the optimized molecules. In the context of compound design X (Y) can be, e.g., the set of inactive (active) molecules.

To represent the sets X and Y our approach requires an embedding of molecules which is reversible, i.e. enables both encoding and decoding of molecules. For this purpose we use the latent space of Junction Tree Variational Autoencoder (JT-VAE) [7] – we represent each molecule as a point in the latent space, given by the mean of the variational encoding distribution [9]. This approach has the advantage that the distance between molecules (required to calculate the loss function) can be defined directly in the latent space.

Our model works as follows: (i) we define the sets X and Y (e.g., inactive/active molecules); (ii) we introduce the mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$; (iii) we introduce discriminator D_X (and D_Y) which forces the generator F (and G) to generate samples from a distribution close to the distribution of X (or Y). The components F , G , D_X , and D_Y are modeled by neural networks (see subsect. 2.1 for technical details).

The main idea is to: (i) take the prior molecule x without a specified feature (e.g. activity) from set X , and compute its latent space embedding; (ii) use the generative neural network G to obtain the embedding of molecule $G(x)$, that has this feature (as if the $G(x)$ molecule came from set Y) but is also similar to the original molecule x ; (iii) decode the latent space coordinates given by $G(x)$ to obtain the optimized molecule. Thereby, the method is applicable in *lead optimization* processes, as the generated compound $G(x)$ remains structurally similar to the input molecule.

To train the Mol-CycleGAN we use the following loss function:

$$L(G, F, D_X, D_Y) = L_{\text{GAN}}(G, D_Y, X, Y) + L_{\text{GAN}}(F, D_X, Y, X) + \lambda_1 L_{\text{cyc}}(G, F) + \lambda_2 L_{\text{identity}}(G, F), \quad (1)$$

and aim to solve

$$G^*, F^* = \arg \min_{G, F} \max_{D_X, D_Y} L(G, F, D_X, D_Y). \quad (2)$$

We use the adversarial loss introduced in LS-GAN [11]:

$$L_{\text{GAN}}(G, D_Y, X, Y) = \frac{1}{2} \mathbb{E}_{y \sim p_{\text{data}}(y)} [(D_Y(y) - 1)^2] + \frac{1}{2} \mathbb{E}_{x \sim p_{\text{data}}(x)} [(D_Y(G(x)))^2], \quad (3)$$

which ensures that the generator G (and F) generates samples from a distribution close to the distribution of Y (or X).

The cycle consistency loss:

$$L_{\text{cyc}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1], \quad (4)$$

reduces the space of possible mapping functions, such that for a molecule x from set X , the GAN cycle brings it back to a molecule similar to x , i.e. $F(G(x))$ is close to x (and analogously $G(F(y))$ is close to y).

Finally, to ensure that the generated (optimized) molecule is close to the starting one, we use the identity mapping loss [19]:

$$L_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|F(y) - y\|_1] + \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|G(x) - x\|_1], \quad (5)$$

which further reduces the space of possible mapping functions and prevents the model from generating molecules that lay far away from the starting molecule in the latent space of JT-VAE.

In our experiments, we use the hyperparameters $\lambda_1 = 0.3$ and $\lambda_2 = 0.1$. Note that these parameters control the balance between *improvement* in the optimized property and *similarity* between the generated and the starting molecule.

2.1 Workflow

We conduct experiments to test if the proposed model is able to generate molecules that are close to the starting ones and possess increased octanol-water partition coefficient (logP) penalized by the synthetic accessibility (SA) score. We optimize penalized logP, while constraining the degree of deviation from the starting molecule. The similarity between molecules is measured with Tanimoto similarity on Morgan Fingerprints [14].

We use the ZINC-250K dataset used in similar studies [7, 10] which contains 250000 drug-like molecules extracted from the ZINC database [16]. The sets X_{train} and Y_{train} are random samples of size 80000 from ZINC-250K, where the compounds' penalized logP values are below and above the median, respectively. X_{test} is a separate, non-overlapping dataset, consisting of 800 molecules with the lowest values of penalized logP in ZINC-250K.

All networks are trained using the Adam optimizer [8] with learning rate 0.0001, batch normalization [6] and leaky-ReLU with $\alpha = 0.1$. The models are trained for 300 epochs. Generators are built of four fully connected residual layers, with 56 units. Discriminators are built of 7 dense layers of the following sizes: 48, 36, 28, 18, 12, 7, 1 units.

Table 1. Results of the constrained optimization for JT-VAE [7], Graph Convolutional Policy Network (GCPN) [18] and Mol-CycleGAN.

δ	JT-VAE		GCPN		Mol-CycleGAN	
	Improvement	Similarity	Improvement	Similarity	Improvement	Similarity
0	1.91 ± 2.04	0.28 ± 0.15	4.20 ± 1.28	0.32 ± 0.12	8.30 ± 1.98	0.16 ± 0.09
0.2	1.68 ± 1.85	0.33 ± 0.13	4.12 ± 1.19	0.34 ± 0.11	5.79 ± 2.35	0.30 ± 0.11
0.4	0.84 ± 1.45	0.51 ± 0.10	2.49 ± 1.30	0.47 ± 0.08	2.89 ± 2.08	0.52 ± 0.10
0.6	0.21 ± 0.75	0.69 ± 0.06	0.79 ± 0.63	0.68 ± 0.08	1.22 ± 1.48	0.69 ± 0.07

3 Results

We optimize the penalized logP under the constraint that the similarity between the original and the generated molecule is higher than a fixed threshold (denoted as δ). This is a realistic scenario in drug discovery, where the development of new drugs usually starts with known molecules such as existing drugs [2].

We maximize the penalized logP coefficient and use the Tanimoto similarity with the Morgan fingerprint to define the threshold of similarity. We compare our results with previous similar studies [7, 18].

In our optimization procedure, each molecule is fed into the generator to obtain the ‘optimized’ molecule $G(x)$. The pair $(x, G(x))$ defines an ‘optimization path’ in the latent space of JT-VAE. To be able to make a comparison with the previous research [7] we start the procedure from the 800 molecules with the lowest values of penalized logP in ZINC-250K and then we decode molecules from 80 points along the path from x to $G(x)$ in equal steps. From the resulting set of molecules we report the molecule with the highest penalized logP score

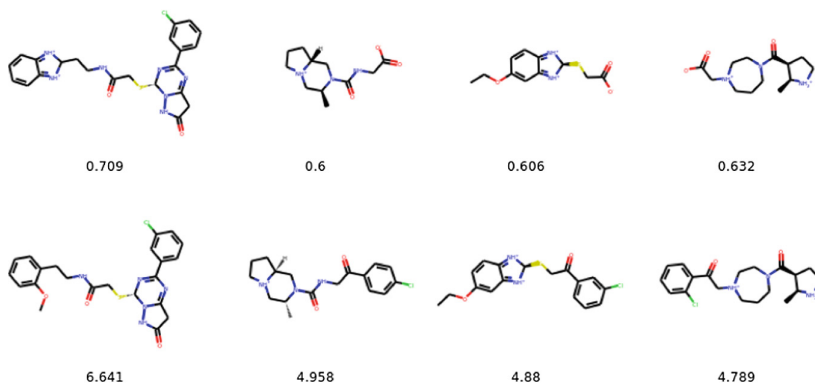


Fig. 1. Molecules with the highest improvement of the penalized logP for $\delta \geq 0.6$. In the top row we show the starting and in the bottom row the optimized molecules. Upper row numbers indicate Tanimoto similarities between the starting and the final molecule. The improvement in the score is given below the generated molecules.

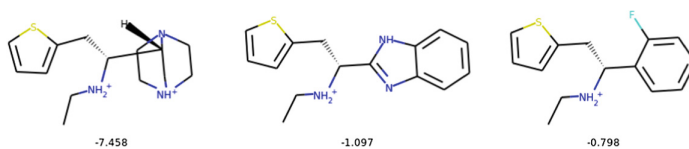


Fig. 2. Evolution of a selected exemplary molecule during constrained optimization. We only include the steps along the path where a change in the molecule is introduced. We show values of penalized logP below the molecules.

that satisfies the similarity constraint. In the task of optimizing penalized logP of *drug-like* molecules, our method significantly outperforms the previous results in the mean improvement of the property (Table 1) and achieves a comparable mean similarity in the constrained scenario (for $\delta > 0$).

Molecules with highest improvement of logP are presented in Fig. 1 with the improvement given below the generated molecules.

Figure 2 shows starting and final molecules, together with all molecules generated along the optimization path and their values of penalized logP.

4 Conclusions

In this work, we introduce Mol-CycleGAN – a new model based on CycleGAN which can be used for the *de novo* generation of molecules. The advantage of the proposed model is the ability to learn transformation rules from the *sets* of compounds with desired and undesired values of the considered property. The model can generate molecules with desired properties, as shown on the example of penalized logP. The generated molecules are close to the starting ones and the degree of similarity can be controlled via a hyperparameter. In the task of constrained optimization of drug-like molecules our model significantly outperforms previous results.

The code used to produce the reported results can be found online at <https://github.com/ardigen/mol-cycle-gan>.

References

1. Bajorath, J.: Integration of virtual and high-throughput screening. *Nat. Rev. Drug Discov.* **1**(11), 882–894 (2002). <https://doi.org/10.1038/nrd941>
2. Besnard, J., et al.: Automated design of ligands to polypharmacological profiles. *Nature* **492**(7428), 215 (2012). <https://doi.org/10.1038/nature11691>
3. Bjerrum, E.J., Threlfall, R.: Molecular generation with recurrent neural networks (RNNs). *arXiv preprint*. [arXiv:1705.04612](https://arxiv.org/abs/1705.04612) (2017)
4. Chen, H., Engkvist, O., Wang, Y., Olivecrona, M., Blaschke, T.: The rise of deep learning in drug discovery. *Drug Discov. Today* **23**(6), 1241–1250 (2018). <https://doi.org/10.1016/j.drudis.2018.01.039>

5. Gupta, A., Müller, A.T., Huisman, B.J., Fuchs, J.A., Schneider, P., Schneider, G.: Generative recurrent networks for de novo drug design. *Mol. Inform.* **37**(1–2), 1700111 (2018). <https://doi.org/10.1002/minf.201700111>
6. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456. ICML 2015, JMLR.org (2015). <http://dl.acm.org/citation.cfm?id=3045118.3045167>
7. Jin, W., Barzilay, R., Jaakkola, T.: Junction tree variational autoencoder for molecular graph generation. In: Dy, J., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 80, pp. 2323–2332. PMLR, Stockholmsmässan, Stockholm (10–15 July 2018)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
9. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint. [arXiv:1312.6114](https://arxiv.org/abs/1312.6114) (2013)
10. Kusner, M.J., Paige, B., Hernández-Lobato, J.M.: Grammar variational autoencoder. In: *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, pp. 1945–1954. JMLR.org (2017)
11. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2794–2802 (2017). <https://doi.org/10.1109/ICCV.2017.304>
12. Rao, V.S., Srinivas, K.: Modern drug discovery process: an in silico approach. *J. Bioinform. Seq. Anal.* **2**(5), 89–94 (2011)
13. Ratti, E., Trist, D.: The continuing evolution of the drug discovery process in the pharmaceutical industry. *Farmaco* **56**(1–2), 13–19 (2001). [https://doi.org/10.1016/S0014-827X\(01\)01019-9](https://doi.org/10.1016/S0014-827X(01)01019-9)
14. Rogers, D., Hahn, M.: Extended-connectivity fingerprints. *J. Chem. Inf. Model.* **50**(5), 742–754 (2010). <https://doi.org/10.1021/ci100050t>
15. Segler, M.H., Kogej, T., Tyrchan, C., Waller, M.P.: Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Cent. Sci.* **4**(1), 120–131 (2017). <https://doi.org/10.1021/acscentsci.7b00512>
16. Sterling, T., Irwin, J.J.: Zinc 15-ligand discovery for everyone. *J. Chem. Inf. Model.* **55**(11), 2324–2337 (2015). <https://doi.org/10.1021/acs.jcim.5b00559>
17. Winter, R., Montanari, F., Noé, F., Clevert, D.A.: Learning continuous and data-driven molecular descriptors by translating equivalent chemical representations. *Chem. Sci.* **10**(6), 1692–1701 (2019). <https://doi.org/10.1039/C8SC04175J>
18. You, J., Liu, B., Ying, Z., Pande, V., Leskovec, J.: Graph convolutional policy network for goal-directed molecular graph generation. In: *Advances in Neural Information Processing Systems*, pp. 6410–6421 (2018)
19. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232 (2017)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

